CLAIMS

What is claimed is:

1. In a computer system including a processor which contains a first set of N-bit data elements loaded into a first register and a second set of N-bit data elements loaded into a second register, a method for providing extended precision in single instruction multiple data (SIMD) arithmetic operations, comprising the steps of:

fetching an arithmetic instruction from a memory unit;

decoding the arithmetic instruction and reading the first vector register and the second vector register;

executing the arithmetic instruction on corresponding N-bit data elements in the first register and second register to produce corresponding resulting elements;

writing the resulting elements into corresponding elements of an accumulator;

transforming the each resulting element in the accumulator into N-bits; and

writing the transformed elements of N-bit width into a third register.

S26 B1

~~2. The method as recited in Claim 1, wherein said decoding step further comprises the steps of:~~

~~selecting an element from the second register; and~~

~~copying the selected element into the other elements in the second register.~~

~~3. The method as recited in Claim 1, wherein said arithmetic~~ instruction is an addition of corresponding vector elements in the first and second vector registers.

4. The method as recited in Claim 1, wherein said arithmetic instruction is a multiplication of corresponding vector elements in the first and second vector registers.

5. The method as recited in Claim 1, wherein said arithmetic instruction is a subtraction of second vector register elements from the first vector register elements.

6. The method as recited in Claim 1, wherein said accumulator is a register having an integer multiples of 64-bit width.

7. The method as recited in Claim 1, wherein said accumulator is a register of 192-bits.

8. The method as recited in Claim 1, wherein said transformation step further comprises the steps of:

scaling the resulting elements in the accumulator by shifting the values in the resulting elements;

rounding the scaled resulting elements in the accumulator; and

clamping the rounded resulting elements.

9. The method as recited in Claim 1, wherein said third register ~~writing step further comprises the steps of:~~

~~reading a portion of the accumulator elements; and~~

writing the portion of the accumulator elements into the corresponding elements of said third register.

10. The method as recited in Claim 9, wherein the portion is either the low third bits or the high third bits of the elements in the accumulator.

11. The method as recited in Claim 1, wherein the values in the resulting elements are wrapped around the representable range of the accumulator elements.

12. The method as recited in Claim 1, wherein the data elements are integers.

13. The method as recited in Claim 1, wherein the first register, the second register, and the third registers are floating point registers.

14. The method as recited in Claim 1, wherein the first register, the second register, and the third register are each 64-bit wide.

15. The method as recited in Claim 1, wherein N is 8.

16. The method as recited in Claim 1, wherein N is 16.

17. The method as recited in Claim 15, wherein the elements in the ~~accumulator are each 24-bit wide.~~

~~18. The method as recited in Claim 16, wherein the elements in the~~ accumulator are each 48 bit wide.

19. The method as recited in Claim 1, wherein said third register writing step further comprises the steps of:

reading a portion of the accumulator elements; and

writing the portion of the accumulator elements into the corresponding elements of said third register.

20. The method as recited in Claim 19, wherein the portion is chosen from the ~~low third bits~~, the middle third bits, or the high third bits of the elements in the accumulator.

21. In a computer system including a processor which contains a first set of N-bit data elements loaded into a ~~first~~ register, a second set of N-bit data elements loaded into a second register, and an accumulator having a third set of data elements, a method for providing extended precision in single instruction multiple data (SIMD) arithmetic operations, comprising the steps of:

fetching an arithmetic instruction from a memory unit;

decoding the arithmetic instruction and reading the first vector register and the second vector register;

executing the arithmetic instruction on corresponding data elements in the first and second vector registers to produce corresponding resulting elements;

adding the resulting elements to the ~~corresponding elements~~ in the ~~accumulator;~~

~~writing the resulting elements into the accumulator;~~

transforming the each resulting element in the accumulator into an N-bit width element; and

writing the transformed elements of N-bit width into a third register.

22. The method as recited in Claim 21, wherein said decoding step further comprises the steps of:

selecting an element from the second register; and

copying the selected element into the other elements in the second register.

23. The method as recited in Claim 21, wherein said arithmetic instruction is an addition of corresponding vector elements in the first and second vector registers.

24. The method as recited in Claim 21, wherein said arithmetic instruction is a multiplication of corresponding vector elements in the first and second vector registers.

25. The method as recited in Claim 21, wherein said arithmetic instruction is a subtraction of second vector register elements from the first vector register elements.

26. The method as recited in Claim 21, wherein said accumulator is ~~a register having an integer multiples of 64-bit width.~~

~~27. The method as recited in Claim 21, wherein said accumulator is a register of 192-bits.~~

28. The method as recited in Claim 21, wherein said transformation step further comprises the steps of:

scaling the resulting elements in the accumulator by shifting the values in the resulting elements;

rounding the scaled resulting elements in the accumulator; and

clamping the rounded resulting elements.

29. The method as recited in Claim 21, wherein said third register writing step further comprises the steps of:

reading a portion of the accumulator elements; and

writing the portion of the accumulator elements into the corresponding elements of said third register.

30. The method as recited in Claim 29, wherein the portion is either the low third bits or high third bits of the elements in the accumulator.

31. The method as recited in Claim 21, wherein the values in the resulting elements are wrapped around the representable range of the accumulator elements.

32. The method as recited in Claim 21, ~~wherein the data elements are integers.~~

33. ~~The method as recited in Claim 21,~~ wherein ~~the first~~ register, the second register, and the third registers are floating point registers.

34. The method as recited in Claim 21, wherein the first register, the second register, and the third register are each 64-bit wide.

35. The method as recited in Claim 21, wherein N is 8.

36. The method as ~~recited~~ in Claim 21, wherein N is 16.

37. The method as recited in Claim 35, wherein the elements in the accumulator are each 24 bit wide.

38. The method as recited in Claim 36, wherein the elements in the accumulator are each 48 bit wide.

39. The method as recited in Claim 21, wherein said third register writing step further comprises the steps of:

reading a portion of the accumulator elements; and

writing the portion of the accumulator elements into the corresponding elements of said third register.

40. The method as recited in Claim 39, wherein the portion is chosen from the low third bits, the middle third bits, or the high third bits of the ~~elements~~ in ~~the accumulator~~.

add B1 > Add D5 >